

AlgoExpert

Quad Layout

Java

14px

Emacs

Monokai

Our Solution(s)

Run Code

Solution 1 Solution 2

```

2
3 ▼:class Program {
4     // O(nm) time | O(min(n, m)) space
5 ▼ public static int levenshteinDistance(String str1, String str2) {
6     String small = str1.length() < str2.length() ? str1 : str2;
7     String big = str1.length() >= str2.length() ? str1 : str2;
8     int[] evenEdits = new int[small.length() + 1];
9     int[] oddEdits = new int[small.length() + 1];
10 ▼ for (int j = 0; j < small.length() + 1; j++) {
11     evenEdits[j] = j;
12 }
13
14     int[] currentEdits;
15     int[] previousEdits;
16 ▼ for (int i = 1; i < big.length() + 1; i++) {
17 ▼     if (i % 2 == 1) {

```

Tests

Your Solutions

Run Code

Solution 1 Solution 2 Solution 3

```

12     }
13
14 ▼ for(r=1;r<str1.length()+1;r++) {
15 ▼     for (c = 1; c < str2.length()+1; c++) {
16 ▼         if(str1.charAt(r-1)==str2.charAt(c-1)) {
17             edits[r][c]=edits[r-1][c-1];
18         }
19         else
20             edits[r][c]=Math.min(Math.min(edits[r-1][c-1], edits[r-1][c]), edits[r][c-1]);
21     }
22 }
23 return edits[str1.length()][str2.length()];
24
25 }
26
27

```

Custom Output

Submit Code



Congratulations!
Your submission passed all of our test cases!